EV316936282

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

Time References For Multimedia Objects

Inventor:

Sean Hayes
Robert Kilroy Hughes

ATTORNEY'S DOCKET NO. MS1-1548US

PRIORITY TO PROVISIONAL APPLICATION

This application claims priority to provisional application serial number 60/416,976 filed October 7, 2002.

TECHNICAL FIELD

The systems and methods described herein relate to associating time with multimedia objects and specifically to systems and methods that provide and/or redefine time references for multimedia objects.

BACKGROUND

Multimedia presentations are made up of discrete multimedia or audio video (A/V) objects (multimedia objects). Multimedia objects may originate from one source such as a digital versatile disc (DVD) disc played on a drive on a personal computer (PC). In other cases multimedia objects may originate from (i.e., streamed from) multiple sources and played at a common destination. For example, multimedia objects may originate from multiple websites and played at a PC. Multimedia presentations may include and combine multimedia objects from videotape, a television broadcast, DVD, and from a website on the Internet (i.e., evolving web broadcast).

Multimedia objects include A/V frames, pictures, video clips, audio clips, and audio samples. Various formats may describe multimedia objects and include static image, animation, video, and text formats. Static image formats include portable network graphics (PNG), joint photographic experts group (JPEG), and motion picture experts group (MPEG). Animation formats include multiple-image network graphics (MNG). Video formats include MPEG video, audio video

interleave (AVI) and various streaming formats. Text formats include extensible markup language (XML), hypertext markup language (HTML), and extensible HTML (XHTML).

Synchronized multimedia integration language (SMIL) is a declarative language for describing structural, spatial and temporal effects such as animation in a presentation. A SMIL listing or document describes when multimedia objects are displayed. In particular, a SMIL document makes use of "time containers" that group data (i.e., multimedia objects). SMIL time containers include a tag <par> which "plays" pieces of data in parallel (i.e., the same time); a tag <seq> which plays data sequentially (i.e. one after the other in the time container list); and a tag <excl> which plays a piece of data exclusive of other data (i.e., no particular sequence).

Although data or media objects may originate from one or multiple sources, when using SMIL, data or media objects must be "linked" together by a single SMIL document. In other words, the single SMIL document is needed to link multimedia objects with one another.

An "element" is a basic unit of an XML document. An element can include other elements and text, and begins with an opening tag <name>, and ends with a closing tag </name>. Tags that begin with <?, and end with ?> are processing instructions that specify particular style sheets to be used. Style sheets are distinct documents that define format for tags of an XML document.

XML tags identify and describe data or multimedia objects that are contained in or referenced by the element. Attributes in the element further describe the purpose and content of data or multimedia elements of the element.

Data that is part of an XML document, in particular an element, is available to other XML documents.

Associated with an XML document are schemas used to describe which tags may be used in which XML documents. Such schemas include tag formatting performed by eXtensible Stylesheet Language for Formatting Objects (XSL-FO) or cascading style sheets.

An XSL-FO document is a transformed XML document that contains information as to structure and appearance.

External style sheets may be provided as a reference for documents as to structure and appearance information. In other words, external style sheets provide the ability to describe appearance and structure of objects in other documents; however, external style sheets and XML documents do have the ability to describe when objects are to be played or presented.

XML does not provide temporal reference for multimedia objects. SMIL provides the temporal reference for multimedia objects; however, SMIL is relatively inflexible and requires that multimedia objects be reference directly by a single SMIL document. In other words, unlike extensible textual markup languages like XML, SMIL does not allow documents to control other documents.

SUMMARY

The systems and methods described herein include referencing multimedia objects by elements in a document, wherein those elements are further associated with other elements which are arranged to indicate a rendition timing.

In certain embodiments an external document is used to provide the rendition timing.

Particular embodiments provide that the external document is made aware of events affecting the document.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram illustrating a multimedia presentation described by a relationship of objects, content documents, a timing document, and events affecting elements in a content document.

Fig. 2 is a flow chart illustrating how multimedia objects are provided a rendition timing.

Fig. 3 is a block diagram illustrating a system of multimedia broadcasters providing multimedia objects and presentations to a network connected to multimedia devices to play the multimedia objects and presentations.

Fig. 4 is a block diagram illustrating a general example of a computer that may be used as a multimedia device to play multimedia objects based on rendition timings.

DETAILED DESCRIPTION

Fig. 1 shows documents and objects that are used to define a multimedia presentation 100. Multimedia presentation 100 includes objects 105. Objects 105 may be one or more multimedia objects which are described by various formats that include static image, animation, video, and text formats. Formats may include PNG, JPEG, MPEG, MNG, AVI, XML, HTML, and XHTML. In this example objects 105 include object1 105(1) and object2 105(2). Although shown as a group and originating from a single source (i.e., objects 105), object1 105(1) and object2 105(2) may originate from different sources.

The multimedia presentation 100 includes document Doc01.xml 110. Doc01.xml is a "content document" that is formatted in a textual markup language such as XML or SGML (HTML). Doc01.xml 110 includes tagged elements that reference multimedia objects. The tagged elements of Doc01.xml 110 are element1 and element2. In this example, the tagged elements of Doc01.xml 110 reference object1 105(1) and object2 105(2). A parent element, element n, may include element1 and element2. Element n is considered one level up from element1 and element2.

An identifier may be used by Doc01.xml 110 to identify element n. Such an identifier may be "ID1". Unique identifiers may also be used to identify element1 and element2.

The elements of Doc01.xml 110 as shown as shown as group 112 in Fig. 1 are listed as follows:

```
<element n>
    <element1>object1</element1>
    <element2>object2</element1>
</element n>
```

In other embodiments element1 and element2 are grouped separately.

Multimedia presentation 100 further includes a document Timing.xml 115. Although Doc01.xml 110 and Timing.xml 115are logically distinct document types, they may be packaged together as a single logical file. Timing.xml 115 is a "timing document" that may also formatted in a textual markup language such as XML. Timing.xml 115 includes a time container 117 that describe the behavior of element1 and element2 of Doc01.xml 110. Time containers provide rendition

timings for elements that may in turn reference multimedia objects. Time container 117 of Timing.xml 115 as shown in Fig. 1 is listed as follows:

```
<par>
        <cue select = "element1" />
        <cue select = "element2" />
</par>
```

The element "cue" is used to target or reference specific elements in a timed document. In this example, "cue" is used to reference element1 and element2 of Doc1.xml 110. Doc1.xml 110 is considered a timed document.

Time container 117 illustrated in Timing.xml 115 includes an element "par" which is used to group elements, and in particular the "cue" elements that reference element1 and element2. A time container with the element "par" provides that elements of the time container are rendered or played at the same time or in parallel with one another. Therefore, in the example listed above, through the use of "cue", element1 and element2 are played or rendered beginning at the same time. Since element1 references object1 105(1), and element2 references objects 105(2), object1 105(1) and object2 (105(2) are played or rendered at the same time.

Time containers may also use elements "seq" and "excl" to group particular elements, and multimedia objects referenced by those particular elements. An example of a time container with the element "seq" is as follows:

```
<seq>
        <cue select = "element1" />
        <cue select = "element2" />
</seq>
```

In the time container above containing the element "seq", element1 is rendered or played before element2 in the sequence they are listed in the time container. In other words, element1 is rendered after element2 in an ordered list. Multimedia objects referenced by element1 will be rendered or played before multimedia objects referenced by element2.

An example of a time container with the element "excl" is as follows:

```
<excl>
        <cue select = "element1" />
        <cue select = "element2" />
</excl>
```

In the time container above containing "excl", element1 may be rendered or played before or after element2. The element "excl" provides that element1 or element2 are rendered or played exclusive of one another. It does not matter which order the elements are listed in the time container. Multimedia objects referenced by element1 are rendered or played exclusive of multimedia objects referenced by element2. The time containers, in particular, may use SMIL conventions, with the addition of the <cue> element. The <cue> element in the timing document Timing.xml 115 may have child elements which may describe actions applied to another element "elementi" in the timed document Doc1.xml 110.

An event may take place that affects elements in the timed document Doc01.xml 110. An event is special object that is created when something occurs such as an input from a user. An example of an event is a user pausing play by activating a "pause button".

Events in Doc01.xml may be handled through "EventListener" document object model (DOM) constructs as defined by the World Wide Web Consortium (W3C). In particular, events may be looked for and handled through a construct defining an element "listener".

In this example an event "pause button" 120 takes place and may affect element1 or element2 of Doc01.xml 110. Doc1.xml 110 may include a listener element 124 that is listed as follows:

```
<listener
        event = "pause button"
        observer = "ID1"
        handler = "stop"
/>
```

The listener element 124 is configured to watch for the event "pause button" 120. "ID1" identifies element n which might be for example, one of element1 or element2. An element may be a "target" which is affected by the event. For example, element1 and/or element2 may be a target element affected by event "pause button" 120. An "observer" element may or may not be a target element, but is made aware of an event that affects elements (i.e., target elements) that are levels below it (i.e., are children elements of a parent observer element). For example, element n may be an observer when element1 and element2 are

targets. The observer attribute in element 124 uses the identifier "ID1" to identify element n.

An attribute "handler" is activated when an event occurs in the target element. In this example, the attribute handler names an application "stop" to be activated. The application "stop" may be a defined instruction within Doc01.xml to instruct elements to pause play.

Since timing document Timing.xml 115 defines timing for timed document Doc01.xml 110, it may be made aware of event "pause button" 120 affecting elements in Doc01.xml 110.

Timing.xml 115 is provided a proxy listener element 126 that is listed as follows:

```
<listener = "proxy"
        event = "pause button"
        observer = pointer to "ID1"
        handler = "pause"
/>
```

Proxy listener 126 is also configured to look for the event "pause button" 120. Instead of an observer attribute that directly identifies element n through identifier "ID1" a pointer to "ID1" is used in an observer attribute of proxy listener element 126. In this example the observer attribute of proxy listener element 126 is extended to be a selector (i.e., using a pointer such as Xpointer in XML). The syntax "ID1" refers to the identifier of an element (i.e., elementn) in Doc01.xml 110, not an element in Timing.xml 115. An attribute handler may be activated when event "pause button" 120 occurs in a target element (e.g., element1 or

element2). In proxy listener 126, the attribute handler names an application "pause" which instructs time container 117 to pause play.

Proxy listener element 126 in the timing document Timing.xml 115 does not require or rely on having the listener element 124 in Doc01.xml. Proxy listener element 126 points directly at the element using the pointer to "ID1".

Doc01.xml 110 may include the behavior of another content document, and specifically elements of the other content document. In this example, element1 and element 2 of Doc01.xml 110 include the behavior of element3 and element4 of Doc02.xml 130.

The elements of Doc02.xml 130 as shown in Fig. 1 are shown conceptually as follows:

```
<element3>element1</element3>
<element4>element2</element4>
```

In this arrangement Doc02.xml may be associated with its own timing document (i.e., a separate Timing2.xml). Both Doc01.xml and Doc02.xml may use the same timing document; however typically this may be a special case of a more general arrangement. Timing document Timing.xml 115 may directly apply to a number of content documents, either directly or indirectly. If timing changes need to be made, they are made to timing document Timing.xml 115 or Timing.xml 115 may be replaced by another timing document that references elements in Doc01.xml 110.

Although Timing.xml 115 is illustrated as an XML document, alternative timing document constructs may apply. For example, a cascading style sheet

(CSS) may be used. CSS is defined by the W3C and is a language for completely describing an XML content document.

Rendition Timing for Multimedia Objects

Fig. 2 shows a process 200 that provides rendition timing to multimedia objects.

At block 205, a first set of elements, which may be in a document written in an extensible textual markup language such as XML document Doc01.xml 110 of Fig. 1, references one or more multimedia objects which may originate from one or more sources. The reference may be performed by particularly calling out the multimedia objects in elements of the document. The reference may also be performed through the use of pointers in the elements directed to the multimedia objects.

At block 210, the first set of elements are associated to a second set of elements which are arranged to indicate timing. The second set of elements may be part of a the first document, where the first document may be an XSL-FO file. In other cases, the second set of elements is in second document such as an external style sheet.

At block 215, the second set of elements are arranged to indicate timing. In particular the arrangement is performed through the use of a time container, such as illustrated in timing document Timing.xml 115 of Fig. 1. As described above, the element "cue" is used in timing document Timing.xml 115 to associate elements in the first document Doc01.xml 110.

At block 220, the multimedia objects are rendered based on timing arrangement of the second set of elements, since the second set of elements are associated with first set of elements which reference the multimedia objects.

Exemplary System

Fig. 3 shows a system 300 of multimedia devices and broadcasters to provide multimedia objects and multimedia presentations. It is contemplated that each multimedia object be identified by a particular name. The name may be located in a header or in a file listing of the multimedia object, and may be read by a multimedia device.

A network 305 includes local area, wide area, private, and public private networks, including the Internet. Hardwired (e.g., cable) and/or wireless networks may make up network 305. Network 305 is designed to provide communications (i.e., multimedia objects) from and to various entities.

A television broadcast point 310 provides multimedia presentation in system 300. Television broadcast point 310 plays multimedia objects from an A/V source 315. Television broadcast point 310 may intersperse multimedia objects from A/V source 315 with multimedia objects such as commercials or promotions from an added content source 320.

Television broadcast point 310 is connected to a satellite dish 325 which uplinks to a satellite 330. Satellite dish 325 transmits signals carrying multimedia objects to satellite 330. A satellite dish 335 downlinks the signals carrying multimedia objects from satellite 130. Network 305 is directly connected to satellite dish 335 to receive the multimedia objects. Alternatively network 305 may be directly connected to television broadcast point 310. Television broadcast point 310 may also transmit radio frequency (RF) signals carrying the multimedia objects through an antenna 340. Such RF signals may be received by various multimedia devices that include televisions.

A multimedia device 345 is a standalone unit not coupled to network 305, but is able receive the multimedia objects from television broadcast point 310. Multimedia device 345 includes an A/V display unit 345-A which may be a television set or monitor with a tuner and A/V inputs and outputs. Multimedia device 345 may further include a multimedia player 345-B such as a DVD player, and a multimedia recorder 345-C. Multimedia device 345 is connected to an antenna 350 that receives RF signals from sources such as antenna 340, where the RF signals carry multimedia objects. Multimedia objects may further be received by multimedia device 345 through a satellite dish 355, where satellite dish 355 receives signals from satellite 330. The multimedia objects and may be recorded by multimedia recorder 345-C.

Broadcast of multimedia objects may also be performed through the Internet. Typically, this is known as a "web broadcast" and provides digital transmission of multimedia objects. A web broadcast point 360 may include an interactive web site accessible through network 305. Web broadcast 360 receives multimedia presentation from an A/V source 365 with interspersed multimedia objects from an added content source 370. Alternatively such objects may come from other sources.

Web broadcast point 360 delivers multimedia objects to network 305 for access by various multimedia devices connected to network 305. Alternatively, web broadcast point 360 may deliver multimedia objects through a satellite as described above.

A multimedia device 375 is coupled to network 305 and receives multimedia objects from various sources including television broadcast point 310 and web broadcast point 360. Multimedia device 375 includes an A/V display unit

375-A which may be a television set or monitor with a tuner and A/V inputs and outputs. Multimedia device 375 may further include a multimedia player 375-B such as a DVD player, and a multimedia recorder 375-C. Multimedia objects and interspersed multimedia objects may be recorded by multimedia recorder 375-C. Multimedia devices and standalone units include WebTV units, and personal video recorder (PVR) units. A personal computer 380 may also be considered a multimedia device.

A web site 385 illustrates a source from which a multimedia device accesses additional content such as updated files that may include timing documents such as Timing.xml 110 of Fig. 1. Broadcast points such as television broadcast point 310 and web broadcast point 360 may stream new multimedia objects to be sequenced with pre-existing content at various multimedia devices.

Exemplary Computer (Multimedia Device) Environment

The subject matter is described in the general context of computer-executable instructions, such as program modules, being executed by one or more multimedia devices such as multimedia devices 345, 375, and 380 of Fig. 3. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Moreover, those skilled in the art will appreciate that the subject matter may be practiced with other computer system configurations, including hand-held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, and the like. In a distributed computer environment, program modules may be located in both local and remote memory storage devices.

Fig. 4 shows a general example of a computer 430 that is used in accordance with the subject matter. Computer 430 is shown as an example of a computer that can perform the functions of a multimedia device. Computer 430 includes one or more processors or processing units 432, a system memory 434, and a bus 436 that couples various system components including the system memory 434 to processors 432.

The bus 436 represents one or more of any of several types of bus structures, including a memory bus or memory controller, a peripheral bus, an accelerated graphics port, and a processor or local bus using any of a variety of bus architectures. The system memory includes read only memory (ROM) 438 and random access memory (RAM) 440. A basic input/output system (BIOS) 442, containing the basic routines that help to transfer information between elements within computer 430, such as during start-up, is stored in ROM 438. Computer 430 further includes a hard disk drive 444 for reading from and writing to a hard disk, not shown, a magnetic disk drive 446 for reading from and writing to a removable magnetic disk 448, and an optical disk drive 450 for reading from or writing to a removable optical disk 452 such as a CD ROM or other optical media. The hard disk drive 444, magnetic disk drive 446, and optical disk drive 450 are connected to the bus 436 by an SCSI interface 454 or some other appropriate interface. The drives and their associated computer-readable media provide nonvolatile storage of computer readable instructions, data structures, program modules and other data for computer 430.

Although the exemplary environment described herein employs a hard disk, a removable magnetic disk 448 and a removable optical disk 452, it should be appreciated by those skilled in the art that other types of computer readable media

which can store data that is accessible by a computer, such as magnetic cassettes, flash memory cards, digital video disks, random access memories (RAMs) read only memories (ROM), and the like, may also be used in the exemplary operating environment.

A number of program modules may be stored on the hard disk, magnetic disk 448, optical disk 452, ROM 438, or RAM 440, including an operating system 458, one or more application programs 460, other program modules 462, and program data 464.

A user may enter commands and information into computer 430 through input devices such as keyboard 466 and pointing device 468. Other input devices (not shown) may include a microphone, joystick, game pad, satellite dish, scanner, or the like. These and other input devices are connected to the processing unit 432 through interface 470 that is coupled to bus 436. Monitor 472 or other type of display device is also connected to bus 436 via an interface, such as video adapter 474.

Computer 430 operates in a networked environment using logical connections to one or more remote computers, such as a remote computer 476. The remote computer 476 may be another personal computer, a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to computer 430, although only a memory storage device 478 has been illustrated in Fig. 4. The logical connections depicted in Fig. 4 include a local area network (LAN) 480 and a wide area network (WAN) 482. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets, and the Internet.

When used in a LAN networking environment, computer 430 is connected to the local network 480 through a network interface or adapter 484. When used in a WAN networking environment, computer 430 typically includes a modem 486 or other means for establishing communications over the wide area network 482, such as the Internet. The modem 486, which may be internal or external, is connected to the bus 436 via a serial port interface 456. In a networked environment, program modules depicted relative to the personal computer 430, or portions thereof, may be stored in the remote memory storage device. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

Generally, the data processors of computer 430 are programmed by means of instructions stored at different times in the various computer-readable storage media of the computer. Programs and operating systems are typically distributed, for example, on floppy disks or CD-ROMs. From there, they are installed or loaded into the secondary memory of a computer. At execution, they are loaded at least partially into the computer's primary electronic memory.

The subject matter described herein includes these and other various types of computer-readable storage media when such media contain instructions or programs for implementing the steps described below in reference to Fig. 4 in conjunction with a microprocessor or other data processor.

The subject matter also includes the computer itself when programmed according to the methods and techniques described below. Furthermore, certain sub-components of the computer may be programmed to perform the functions and steps described below. The subject matter includes such sub-components when they are programmed as described. In addition, the subject matter described

herein includes data structures, described below, as embodied on various types of memory media.

For purposes of illustration, data, programs and other executable program components, such as the operating system are illustrated herein as discrete blocks, although it is recognized that such programs and components reside at various times in different storage components of the computer, and are executed by the data processor(s) of the computer.

Although the invention has been described in language specific to structural features and/or methodological acts, it is to be understood that the invention defined in the appended claims is not necessarily limited to the specific features or acts described. Rather, the specific features and acts are disclosed as exemplary forms of implementing the claimed invention.